CHAPTER

3 SYNTAX ANALYSIS

Context Free Grammar

1. [MCQ] [GATE-2017 : 2M]

Consider the following expression grammar G:

$E \rightarrow E - T \mid T$

$T \rightarrow T + F \mid F$

$F \rightarrow (E) \mid id$

Which of the following grammars is not left recursive, but is equivalent to G?

(a) $E \rightarrow E{-}T \mid T$
$T \rightarrow T{+}F \mid F$
$F \rightarrow (E) \mid id$

(b) $E \rightarrow TE'$
$E \rightarrow -TE' \mid \varepsilon$
$T \rightarrow T{+}F \mid F$

(c) $E \rightarrow TX$
$X \rightarrow -TX \mid \varepsilon$
$T \rightarrow FY$
$Y \rightarrow +FY \mid \varepsilon$
$F \rightarrow (E) \mid id$

(d) $E \rightarrow TX \mid (TX)$
$X \rightarrow -TX \mid +TX \mid \varepsilon$
$T \rightarrow id$

Top Down Parser

2. [MSQ] [GATE-2025 : 2M]

Which of the following statement(s) is/are TRUE while computing First and Follow during top-down parsing by a compiler?

(a) For a production $A \rightarrow \epsilon$, ϵ will be added to $First(A)$.

(b) If there is any input right end marker, it will be added to $First(S)$, where S is the start symbol.

(c) For a production $A \rightarrow \epsilon$, ϵ will be added to $Follow(A)$.

(d) If there is any input right end marker, it will be added to $Follow(S)$, where S is the start symbol.

3. [MCQ] [GATE-2021 : 2M]

Consider the following context-free grammar where the set of terminals is {a, b, c, d, f}

$S \rightarrow d\ a\ T \mid R\ f$

$T \rightarrow a\ S \mid b\ a\ T \mid \varepsilon$

$R \rightarrow c\ a\ T\ R \mid \varepsilon$

The following is a partially-filled LL(1) parsing table

	a	b	c	d	f	$
S			①	S → da T	②	
T	T→aS	T→baT	③		T→ϵ	④
R			R→caTR		R→ϵ	

Which one of the following choices represents the correct combinations for the numbered cells in the parsing table ('blank' denotes that the corresponding cell is empty)?

(a) (1) S → R f
(2) S → R f
(3) T → ε
(4) T → ε

(b) (1) blank
(2) S → R f
(3) T → ε
(4) T → ε

(c) (1) S → R f
(2) blank
(3) blank
(4) T → ε

(d) (1) blank
(2) S → R f
(3) blank
(4) blank

4. [NAT] [GATE-2019 : 1M]

Consider the grammar given below:

$S \rightarrow Aa$

$A \rightarrow BD$

$B \rightarrow b \mid \varepsilon$

$D \rightarrow d \mid \varepsilon$

Let a, b, d, and $ be index as follows:

A	b	d	$
3	2	1	0

Compute the FOLLOW set of the non-terminal B and write the index values for the symbols in the FOLLOW set in the descending order. (For example, if the FOLLOW set is {a, b, d, $}, then the answer should be 3210)

5. [MCQ] [GATE-2017 : 1M]

Consider the following grammar:

$P \rightarrow xQRS$

$Q \rightarrow yz \mid z$

$R \rightarrow w \mid \varepsilon$

$S \rightarrow y$

What is FOLLOW(Q)?

(a) {R} (b) {w}

(c) {w,y} (d) {w,$}

(Common Data for next 2 questions)

6. [MCQ] [GATE-2012 : 2M]

For the grammar below, a partial LL(1) parsing table is also presented along with the grammar. Entries that need to be filled are indicated as E1, E2 and E3. ε is the empty string, $ indicates end of input, and | separates alternate right hand sides of productions.

$S \rightarrow aAbB \mid bAaB \mid \varepsilon$

$A \rightarrow S$

$B \rightarrow S$

	a	b	$
S	E1	E2	$S \rightarrow \varepsilon$
A	$A \rightarrow S$	$A \rightarrow S$	Error
B	$B \rightarrow S$	$B \rightarrow S$	E3

The FIRST and FOLLOW sets for the non-terminals A and B are

(a) FIRST (A)={a, b, ε}=FIRST (B)
FOLLOW (A)={a, b}
FOLLOW (B)={a, b, $}

(b) FIRST (A)={a, b, $}
FIRST (B)={a, b, ε}
FOLLOW (A)={a,b}
FOLLOW (B)={$}

(c) FIRST (A)={a, b, ε}=FIRST (B),
FOLLOW (A)={a,b}
FOLLOW (B)=$

(d) FIRST (A)={a,b}=FIRST (B)
FOLLOW (A)={a, b}
FOLLOW (B)={a, b}

7. [MCQ] [GATE-2012 : 2M]

The appropriate entries for E1, E2 and E3 are

(a) E1: $S \rightarrow aAbB$, $A \rightarrow S$
E2: $S \rightarrow bAaB$, $B \rightarrow S$
E3: $B \rightarrow S$

(b) E1: $S \rightarrow aAbB$, $S \rightarrow \varepsilon$
E2: $S \rightarrow bAaB$, $S \rightarrow \varepsilon$
E3: $S \rightarrow \varepsilon$

(c) E1: $S \rightarrow aAbB$, $S \rightarrow \varepsilon$
E2: $S \rightarrow bAaB$, $S \rightarrow \varepsilon$
E3: $B \rightarrow S$

(d) E1: $A \rightarrow S$, $S \rightarrow \varepsilon$
E2: $B \rightarrow S$, $S \rightarrow \varepsilon$
E3: $B \rightarrow S$

LR Parser

8. [MCQ] [GATE-2025 : 2M]

Given a Context-Free Grammar G as follows:

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow d$

Which ONE of the following statements is TRUE?

(a) G is neither LALR(1) nor SLR(1)

(b) G is CLR(1), not LALR(1)

(c) G is LALR(1), not SLR(1)

(d) G is LALR(1), also SLR(1)

9. [MSQ] [GATE-2025 : 2M]

Consider two grammars G_1 and G_2 with the production rules given below:

G_1: $S \rightarrow$ if E then S | if E then S else S | a

$E \rightarrow b$

G_2: $S \rightarrow$ if E then S|M

$M \rightarrow$ if E then M else S|c

$E \rightarrow b$

Where if, then else, a, b, c, are the terminals.

Which of the following option(s) is/are CORRECT?

(a) G_1 is not LL(1) and G_2 is LL(1).

(b) G_1 is LL(1) and G_2 is not LL(1)

(c) G_1 and G_2 are not LL(1).

(d) G_1 and G_2 are ambiguous.

10. [NAT] [GATE-2024 : 2M]

Consider the following augmented grammar, which is to be parsed with a SLR parser. The set of terminals is {a, b, c, d, #, @}

$S' \rightarrow S$

$S \rightarrow SS \mid Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d\#$

$B \rightarrow @$

Let I_o = CLOSURE({S' → · S)). The number of items in the set GOTO(Io, S) is _____.

11. [MCQ] [GATE-2024 : 2M]

Consider the following context-free grammar where the start symbol is S and the set of terminals is {a,b,c,d}.

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow cS \mid \in$

$B \rightarrow dS \mid \in$

The following is a partially-filled LL(1) parsing table.

	a	b	c	d	$
S	S→AaAb	S→BbBa	(1)	(2)	
A	A → ∈	(3)	A→cS		
B	(4)	B → ∈		B →dS	

Which one of the following options represents the CORRECT combination for the numbered cells in the parsing table?

Note: In the options, "blank" denotes that the corresponding cell is empty.

(a) (1) S → BbBa (2) S → AaAb (3) blank (4) blank

(b) (1) S → AaAb (2) S → BbBa (3) blank (4) blank

(c) (1) S → BbBa (2) S → AaAb (3) A → ∈ (4) B → ∈

(d) (1) S → AaAb (2) S → BbBa (3) A → ∈ (4) B → ∈

12. [MCQ] [GATE-2024 : 2M]

Consider the following grammar G, with S as the start symbol. The grammar G has three incomplete productions denoted by (1), (2), and (3).

$S \rightarrow daT \mid \underline{(1)}$

$T \rightarrow aS \mid bT \mid \underline{(2)}$

$R \rightarrow \underline{(3)} \mid \in$

The set of terminals is {a, b, c, d, f}. The FIRST and FOLLOW sets of the different non-terminals are as follows.

FIRST(S) = {c, d, f}, FIRST(T) = {a, b, ∈), FIRST(R) = {C, Ɛ}

FOLLOW(S) = FOLLOW(T) = {c, f, $}, FOLLOW(R) = {f}

Which one of the following options CORRECTLY fills in the incomplete productions?

(a) (1) S → Rf (2) T → ∈ (3) R → cTR

(b) (1) S → fR (2) T → cT (3) R → cR

(c) (1) S → fR (2) T → ∈ (3) R → cTR

(d) (1) S → Rf (2) T → cT (3) R → cR

13. [MCQ] [GATE-2024 : 1M]

Which of the following is/are Bottom-Up Parser(s)?

(a) LL(1) Parser

(b) Predictive Parser

(c) Shift-reduce Parser

(d) LR Parser

14. [MCQ] [GATE-2022 : 1M]

Which one of the following statements is TRUE?

(a) The LALR(1) parser for a grammar G cannot have reduce – reduce conflict if the LR(1) parser for G does not have reduce – reduce conflict.

(b) Symbol table is accessed only during the lexical analysis phase.

(c) Data flow analysis is necessary for run-time memory management.

(d) LR(1) parsing is sufficient for deterministic context – free languages.

15. [NAT] [GATE-2021 : 1M]

Consider the augmented grammar with {+, *, (,), id} as the set of terminals.

S′ → S

S → S + R | R

R → R * P | P

P → (S) | id

If I_0 is the set of two $LR(0)$ items {[S→S.], [S →S. + R]}, then $goto(closure(I_0), +)$ contains exactly _____ items.

16. [MCQ] [GATE-2021 : 1M]

Consider the following statements

S_1: Every SLR (1) grammar is unambiguous but there are certain unambiguous grammars that are not SLR(1).

S_2: For any context-free grammar, there is a parser that takes at most $O(n^3)$ time to parse a string of length n.

Which one of the following options is correct?

(a) S_1 is true and S_2 is false

(b) S_1 is false and S_2 is true

(c) S_1 is true and S_2 is true

(d) S_1 is false and S_2 is false

17. [NAT] [GATE-2021 : 2M]

Consider the following augmented grammar with {#, @,<, >,a, b, c} as the set of terminals

S′ →S

S → S # cS

S → SS

S → S@

S → <S>

S → a

S → b

S → c

Let I_0 = CLOSURE({S′ →"." S}). The number of items in the set GOTO(GOTO(I_0,<),<) is _____

18. [NAT] [GATE-2021 : 2M]

Consider the following grammar:

S → aSB | d

B → b

The number of reduction steps taken by a bottom-up parser while accepting the string aaadbbb is _____.

19. [MCQ] **[GATE-2019 : 1M]**

Which one of the following kinds of derivation is used by LR parsers?

(a) Leftmost
(b) Leftmost in reverse
(c) Rightmost
(d) Rightmost in reverse

20. [MCQ] **[GATE-2017 : 1M]**

Which of the following statements about parser is/are CORRECT?

I. Canonical LR is more powerful than SLR.
II. SLR is more powerful than LALR.
III. SLR is more powerful than Canonical LR.

(a) I only
(b) II only
(c) III only
(d) II and III only

21. [MCQ] **[GATE-2015 : 2M]**

Consider the following grammar G.

$S \rightarrow F \mid H$
$F \rightarrow p \mid c$
$H \rightarrow d \mid c$

Where S, F and H are non-terminal symbols, p, d and c are terminal symbols. Which of the following statements(s) is/are correct?

S_1: LL(1) can parse all strings that are generated using grammar G.
S_2: LR(1) can parse all strings that are generated using grammar G.

(a) Only S_1
(b) Only S_2
(c) Both S_1 and S_2
(d) Neither S_1 and S_2

22. [MCQ] **[GATE-2015 : 1M]**

Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?

(a) SLR, LALR
(b) Canonical LR, LALR
(c) SLR, canonical LR
(d) LALR, canonical LR

23. [MCQ] **[GATE-2015 : 1M]**

Which one of the following is TRUE at any valid state in shift-reduce parsing?

(a) Viable prefixes appear only at the bottom of the stack and not inside
(b) Viable prefixes appear only at the top of the stack and not inside
(c) The stack contains only a set of viable prefixes
(d) The stack never contains viable prefixes

24. [MCQ] **[GATE-2014 : 1M]**

A canonical set of items is given below

$S \rightarrow L. > R$
$Q \rightarrow R.$

On input symbol > the set has

(a) a shift-reduce conflict and a reduce-reduce conflict.
(b) a shift-reduce conflict but not a reduce-reduce conflict.
(c) a reduce-reduce conflict but not a shift-reduce conflict.
(d) neither a shift-reduce nor a reduce-reduce conflict.

25. [MCQ] **[GATE-2015 : 2M]**

Consider the following two sets of LR(1) items of LR(1) grammar.

$X \rightarrow c.X, c/d$	$X \rightarrow c.X, \$$
$X \rightarrow .cX, c/d$	$X \rightarrow .cX, \$$
$X \rightarrow .d, c/d$	$X \rightarrow .d, \$$

Which of the following statement related to merging of the two sets in the corresponding LALR parser is/are FALSE?

1. Cannot be merged since look aheads are different.
2. Can be merged but will result in S-R conflict.
3. Can be merged but will result in R-R conflict.
4. Cannot be merged since goto on c will lead to two different sets

(a) 1 only
(b) 2 only
(c) 1 and 4 only
(d) 1, 2, 3 and 4

26. [MCQ] [GATE-2013 : 1M]

What is the maximum number of reduces moves that can be taken by a bottom up parser for a grammar with no epsilon and unit productions (i.e. of type $A \to \varepsilon$ and $A \to a$) to parse a string with n tokens?

(a) $n/2$ (b) $n-1$

(c) $2n-1$ (d) 2^n

27. [MCQ] [GATE-2010 : 2M]

The grammar $S \to aSa \mid bS \mid c$ is

(a) LL(1) but not LR(1)

(b) LR(1) but not LL(1)

(c) Both LL(1) and LR(1)

(d) Neither LL(1) nor LR(1)

28. [MCQ] [GATE-2008 : 2M]

An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if

(a) The SLR(1) parser for G has S-R conflict

(b) The LR(1) parser for G has S-R conflicts

(c) The LR(0) parser for G has S-R conflicts

(d) The LALR(1) parser for G has reduce-reduce conflicts.

29. [MCQ] [GATE-2008 : 1M]

Which of the following describes a handle (as applicable to LR-parsing) appropriately?

(a) It is the position in a sentential form where the next shift or reduce operation will occur.

(b) It is non-terminal whose production will be used for reduction in the next step

(c) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.

(d) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right-hand side of the production may be found.

30. [NAT] [GATE-2019 : 2M]

Consider the augmented grammar given below:

$S' \to S$

$S \to \langle L \rangle \mid id$

$L \to L, S \mid S$

Let I_0 = CLOSURE ({[S' → ·S]}). The number of items in the set GOTO (I_0, ⟨) is: ________.

Operator Precedence Parsing

31. [MCQ] [GATE-2014 : 1M]

Consider the grammar defined by the following production rules, with two operator * and +

$S \to T * P$

$T \to U \mid T * U$

$P \to Q + P \mid Q$

$Q \to Id$

$U \to Id$

Which one of the following is TRUE?

(a) + is left associative, while * is right associativ

(b) + is right associative, while * is left associative

(c) Both + and * are right associative

(d) Both + and * are left associative

32. [NAT] [GATE-2016 : 2M]

The attributes of three arithmetic operators in some programming language are given below.

Operator	Precedence	Associativity	Arity
+	High	Left	Binary
−	Medium	Right	Binary
*	Low	Left	Binary

The value of the expression **2 − 5 + 1 − 7 * 3** in this language is ________.

□□□

ANSWER KEY

1.	(c)	2.	(a, d)	3.	(b)	4.	(31 to 31)
5.	(c)	6.	(a)	7.	(c)	8.	(c)
9.	(c, d)	10.	(9)	11.	(d)	12.	(a)
13.	(c, d)	14.	(d)	15.	(5 to 5)	16.	(c)
17.	(8 to 8)	18.	(7 to 7)	19.	(d)	20.	(a)
21.	(d)	22.	(c)	23.	(c)	24.	(d)
25.	(d)	26.	(b)	27.	(c)	28.	(b)
29.	(d)	30.	(5 to 5)	31.	(b)	32.	(9 to 9)

SOLUTIONS

1. (c)

Eliminating left recursion from the given expression grammar:

I. $E \rightarrow E - T \mid T$ will be converted:

$$\boxed{\begin{array}{l} E \rightarrow TX \\ X \rightarrow -TX \mid \varepsilon \end{array}}$$

II. $T \rightarrow T + F \mid F$ will be converted:

$$\boxed{\begin{array}{l} T \rightarrow FY \\ Y \rightarrow +FY \mid \varepsilon \end{array}}$$

III. $\boxed{F \rightarrow (E) \mid id}$ is same as it is already free from left recursion

The resultant grammar after removing left-recursion is

$E \rightarrow TX$

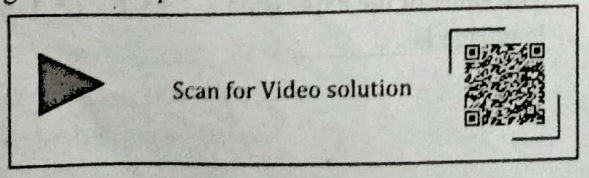
$X \rightarrow -TX \mid \varepsilon$

$T \rightarrow FY$

$Y \rightarrow +FY \mid \varepsilon$

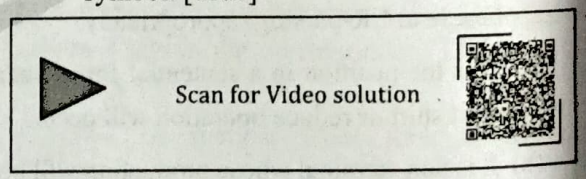
$F \rightarrow (E) \mid id$

The given expression grammar is equivalent to grammar in option C.



2. (a, d)

- For a production $A \rightarrow \epsilon$, ϵ will be added to $First(A)$. [True]
- If there is any input right end marker, it will be added to $First(S)$, where S is the start symbol. [False]
- For a production $A \rightarrow \epsilon$, ϵ will be added to $Follow(A)$. [False]
- If there is any input right end marker, it will be added to $Follow(S)$, where S is the start symbol. [True]



3. (b)

To find out correct combinations for the numbered cells in the parsing table, we need to first find out FIRST and FOLLOW of S, T, and R.

FIRST(S) = {d, c, f}

FOLLOW(S) = {c, f, $}

FIRST(T) = {a, b, e}

FOLLOW(T) = {$, c, f}

FIRST(R) = {c, ε }

FOLLOW(R) = {f}

So,

1. S → Rf [This production can produce c]
2. S → Rf [This production can produce f]
3. T → ε
4. T → ε

So, option A is the correct answer

Scan for Video solution

4. (31 to 31)

Sol- Follow(B) = {d, a}

Index for d is 3

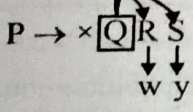
Index for a is 1

So, 31 is the correct answer.

Scan for Video solution

5. (c)

Follow of Q: It depends on First of RS



Follow(Q) = {w, y}

Option c is the correct answer.

Scan for Video solution

6. (a)

First (S) = {a, b, e}

Follow(S) = {$, a, b}

First(A) = {a, b, ε}

First (B) = {a, b, ε}

Follow (A) = {a, b}

Follow (B) = {$, a, b}

So, option A is the correct answer.

Scan for Video solution

7. (c)

Given,

S → aAbB | bAaB | ε

A → S

B → S

	a	b	$
S	E1	E2	S → ε
A	A→ S	A→ S	Error
B	B→ S	B→ S	E3

First (S) = {a, b, ε }

Follow(S) = {$, a, b}

E_1 : {S → a AbB, S → ε }

E_2 : {S → bAaB, S → ε }

E_3: {B → S}

Therefore, option C is the correct answer.

Scan for Video solution

8. (c)

S → Aa|bAc|dc|bda

A → d

Given CFG is CLR(1) and also LALR(1) but not SLR(1).

Scan for Video solution

9. (c, d)

G_1: S → if E then S | if E then S else S|a

Common prefix if not LL(1)

G_2: G_2 is also not LL(1) common prefix if .

Both G_1 and G_2 are ambiguous for string "if E then if E then a else a".

Scan for Video solution

10. (9)

S'→ .S	→S	S'→ S.
S'→ .SS		S → S.S

S'→ .Aa	S → .SS
S'→ .bAc	S → .Aa
S'→ .Bc	S → .bAc
S'→ .bBa	S → .Bc
A→ .d#	S → .bBa
S'→ .@	S → .d#
I_0 state	S → .@

Number of items = 9

Scan for Video solution

11. (d)

Follow (A) = {a, b}

Correct sequence =

(1) S → AaAb (2) S → BbBa (3) A → ∈ (4) B → ∈

Scan for Video solution

12. (a)

(1) S → Rf (2) T → ∈ (3) R → cTR

Scan for Video solution

13. (c, d)

- Shift-reduce Parser
- LR Parser

Scan for Video solution

14. (d)

(a) **FALSE.** The LALR(1) parser for a grammar G may have reduce – reduce conflict if the LR(1) parser for G does not have reduce – reduce conflict.

(b) **FALSE.** Symbol table can be accessed in all phases of compiler.

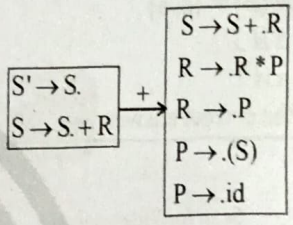
(c) **FALSE.** Data flow analysis is required to imporve the code but not necessary for run-time memory mangement.

(d) **TRUE.** Every LR(1) CFG going to generate DCFL.

Hence, option d is the correct answer.

Scan for Video solution

15. (5 to 5)

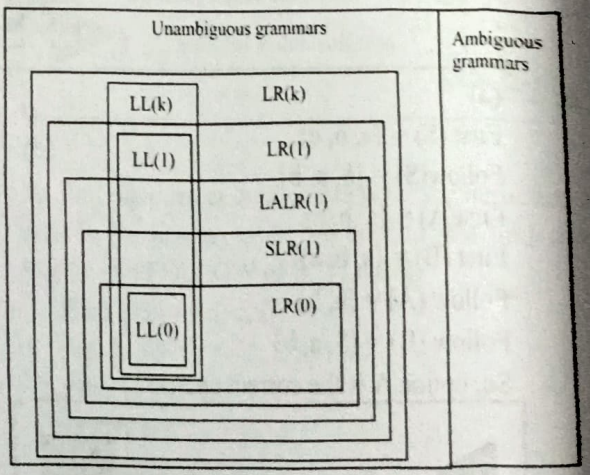


So, there are total 5 items in the goto(closure(I_0), +).

Scan for Video solution

16. (c)

S_1: Every SLR (1) grammar is unambiguous but there are certain unambiguous grammars that are not SLR(1). **TRUE.** For example, LALR(1), CLR(1) are certain unambiguous grammars which are not SLR(1).



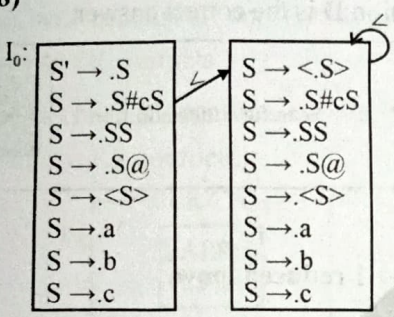
S_2: For any context-free grammar, there is a parser that takes at most $O(n^3)$ time to parse a string of

length n. **TRUE.** For example, CYK algorithm takes $O(n^3)$ time to parse a string of length n.

So, option c is the correct answer.

Scan for Video solution

17. (8 to 8)



There are 8 items in the set GOTO(GOTO(I_0,<),<).

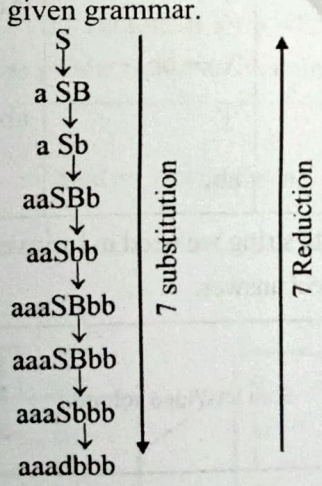
Scan for Video solution

18. (7 to 7)

Bottom-up parser follows reverse of RMD (Right most derivation).

And, Number of reduction steps in reverse of RMD = Number of substitutions in RMD

RMD for given grammar.



So, total 7 number of reduction steps taken by bottom-up parser while verifying the string aaadbbb.

Scan for Video solution

19. (d)

LR parser also known as bottom-up parser and bottom-up parser uses rightmost derivation in reverse.

Hence, option d is the correct answer.

Scan for Video solution

20. (a)

SLR is less powerful than LALR and LALR is less powerful than CLR.

SLR < LALR < CLR.

I. Canonical LR is more powerful than SLR. **TRUE**

II. SLR is more powerful than LALR. **FALSE**

III. SLR is more powerful than Canonical LR. **FALSE**

Statement I is only correct, option a is the right answer.

Scan for Video solution

21. (d)

If we want to drive the string "c", it can be done in two ways S → F→c or S → H → c

So, we can say the given grammar is ambiguous. With the ambiguous grammar, parser cannot parse the string. So, in case of LL(1) or LR(1) parser is going to have a problem, that's why both the statements are incorrect. Hence, option D is the right answer.

Scan for Video solution

22. (c)

Easy to implement, Most powerful.

SLR is easy to implement and CLR is most powerful. So SLR, CLR, is the correct answer.

Scan for Video solution

23. (c)

(a) INCORRECT. Viable prefixes can be present anywhere in the stack.

(b) INCORRECT. Viable prefixes can be present anywhere in the stack.

(c) CORRECT. The stack contains only a set of viable prefixes.

(d) INCORRECT. The stack contains viable prefixes.

Scan for Video solution

24. (d)

$S \rightarrow L.>R$
$Q \rightarrow R.$
$>$
$S \rightarrow L>.R$

This state does not have any conflict; therefore, option D is the correct answer.

Scan for Video solution

25. (d)

There is no conflicts in the given sets and there are no reduced items.

So,

1. "Cannot be merged since look aheads are different" is FALSE. Because they can be merged when lookaheads are different.
2. "Can be merged but will result in S-R conflict" is FALSE. Because no conflict present.
3. "Can be merged but will result in R-R conflict" is FALSE. Because no conflict when merged.

   $X \rightarrow C.X, C/d/\$$

   $X \rightarrow .CX, C/d/\$$

   $X \rightarrow .d, C/d/\$$
4. "Cannot be merged since goto on c will lead to two different sets" FALSE. While merging we do not look at goto.

So, option D is the correct answer.

Scan for Video solution

26. (b)

n = 2 ⇒ 1 reduced move

n = 2 ⇒ 2 reduced moves

n = 4 ⇒ 3 reduced moves

w = ab	w = abc	w = abcd
S	S	S
↓	↓	↓
ab	S → a ×	AB
	ax	↓
	↓	Acd
	X → bc	↓
	↓	abcd
	abc	

For n length string we need n-1 moves. So, option B is the correct answer.

Scan for Video solution

27. (c)

Given,

$S \rightarrow aSa \mid bS \mid c$

First check for LL(1) because Every LL(1) is LR(1)

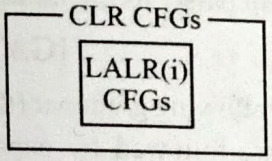
The given grammar is LL(1). So, option C is the correct answer.

Scan for Video solution

28. **(b)**

LALR parser uses the LR(1) items of CLR parser. LALR will contain SR conflicts if and only if LR parser has SR conflicts.

LALR can have RR conflicts even though LR parser do not have RR conflicts.



So, option B is correct answer

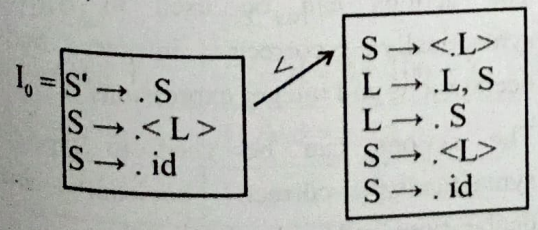
Scan for Video solution

29. **(d)**

In a production whole RHS part is known as handle. It can be defined as the production p that will be used for reduction in the next step along with a position in the sentential form where the right-hand side of the production may be found.

Scan for Video solution

30. **(5 to 5)**

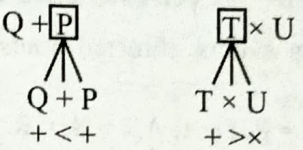


There are total 5 items in in the set GOTO (I_0, ⟨).

Scan for Video solution

31. **(b)**

Operator < Operator	Operator > Operator
Right associate	Left associate



+ is right associative

* is left associative

So, option B is correct.

Scan for Video solution

32. **(9 to 9)**

Given,

+ has highest prcedence and left associative.

– has medium precedence and right associative.

* has lowest precedence and left associative.

Expression = 2 – 5 + 1 – 7 * 3

= 2- [5+1] – 7 * 3

= 2 – 6 – 7 * 3

= [2- [6 - 7]]*3

= [2-(-1)] * 3

= 3 * 3

= 9

So, 9 is the correct answer.

Scan for Video solution